Project 2 algorithm

1. At the start of the program, declare two integer constants for rows and columns of the array and initialize both with the value of 10. (const int rows = 10;)
2. Then open your main function and declare two 2D arrays of integers using the constants in the first step. (int num\_array[rows][cols];­) one called num\_ array will not be shown to the user but another one called disp\_array.
3. Ask the user for their email address with a suitable cout statement: (“Please input your email:”). Using about four to five for loops check to see if the email is in the right format in a different Boolean function called isWrongformat:

“A valid email address must have only one @ (at), only one. (dot) and the remaining characters are only alphanumeric characters (i.e., a-z, A-Z, and 0-9). ‘@’ (at) must come before ‘.’ (dot) and there must be at least one alphanumeric character between them. The function must return true if the format is incorrect, otherwise, return false.” For example:

* Each for loop will have at least one if statement to check at least one of the above parameters for a valid email address.

(For int = i; i<str1.size(); ++i) {

If (str1[i] == ‘.’) {

countdot++;

If (countdot>1) {

Cout<< “Email invalid.” <<endl;

Return true;

}}

Return false;

}

1. Note that the constants declared earlier are the bounds of the 2D array. Using a suitable cout statement ask the user for the row and column number (cout<< “Please input number of rows” <<endl;) but if it exceeds 10 ask them to input it again and not to let it pass 10.

* You can do this by applying if statements after the cin>> statement to make sure it does not pass 10.

If (userinput>10) {

Cout<< “Your input is greater than 10. Please enter another number.” <<endl.

}

* You can put it in a loop with a parameter that requires it to continue unless the userinput<10.

1. Write a function named initialize that initializes the two arrays. The function will have an integer parameter whose default value is -1. [initialize (disp\_array, -1);]

* If the value of -1 is overridden, using the rand () function assign a random number between 1 and 20 to the array. (Rand () %20 +1;)
* If not, then all other elements in the array should be assigned with a value of -1.
* The same function will be used to initialize both arrays. Remember, your function call will determine which array you are initializing. If you override the default value you are initializing num\_array, if you use the default value, you are initializing disp\_array.

1. You will write another separate function called to display to show the 2D array of integers every time it needs to be displayed. (int display( disp\_array []).
2. Initialize an integer variable named points with 10. This is the balance of points with the user. (int points = 10;)